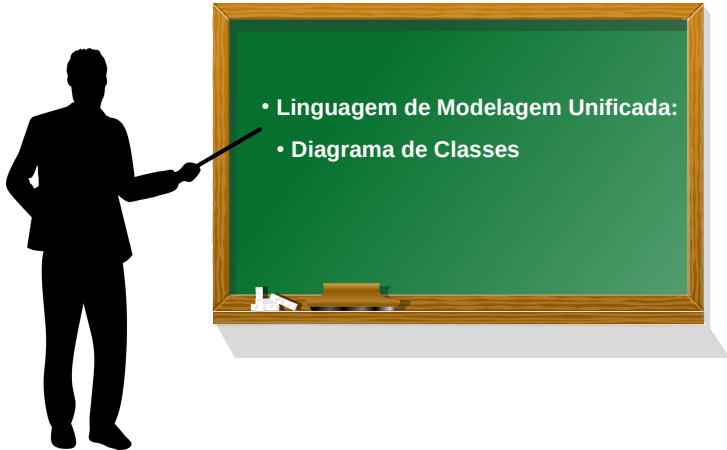


Diagrama de Classes UML

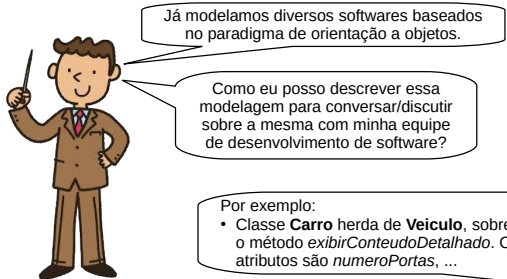
Luiz Henrique de Campos Merschmann
Departamento de Computação Aplicada
Universidade Federal de Lavras

luiz.hcm@ufla.br

Na Aula de Hoje



Representação Textual x Gráfica



Já modelamos diversos softwares baseados no paradigma de orientação a objetos.


Como eu posso descrever essa modelagem para conversar/discutir sobre a mesma com minha equipe de desenvolvimento de software?

Isso é muito simples! Basta você gerar um texto que descreva a sua modelagem.


Por exemplo:

- Classe **Carro** herda de **Veiculo**, sobrescrevendo o método *exibirConteudoDetalhado*. Os seus atributos são *numeroPortas*, ...

Será que não existe uma maneira mais fácil do que essa descrição textual?



Eu adoro figuras! Acho que uma representação gráfica seria bem melhor!



Linguagem de Modelagem Unificada



Certamente a representação gráfica é mais interessante do que a textual, uma vez que nos permite visualizar rapidamente as diversas características do sistema que será implementado.



A UML (*Unified Modeling Language*) é uma linguagem visual utilizada para modelar softwares que utilizam o paradigma de Programação Orientada a Objetos.

A UML se tornou a linguagem **padrão** de modelagem adotada **internacionalmente** pela **indústria** de engenharia de **software**.

Portanto, para quem trabalha direta ou indiretamente com desenvolvimento de software, é importante conhecer UML.



Nesta disciplina apresentaremos **apenas um** dos muitos **diagramas** oferecidos pela UML, o **Diagrama de Classes**. Ele é provavelmente o mais utilizado e um dos mais importantes da UML.

Diagrama de Classes

Seu principal objetivo é permitir a visualização das classes que comporão o sistema.

O diagrama de classes apresenta basicamente:

- ▶ A **estrutura** de uma classe.
 - ▶ Seu **nome**, seus **atributos** e **métodos**.
 - ▶ **Relacionamento** entre as classes.

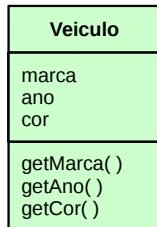
O diagrama de classes apresenta uma visão estática de como as classes estão organizadas, definindo a estrutura lógica das mesmas.

Representação de uma Classe

Como uma classe é representada no Diagrama de Classes?

Uma classe é representada com um retângulo com até três divisões:

- ▶ A primeira contém a descrição ou o nome da classe.
- ▶ A segunda apresenta os atributos da classe.
- ▶ A terceira lista os métodos da classe.



Visibilidade dos Atributos e Métodos

A visibilidade é utilizada para indicar o nível de acessibilidade de um determinado atributo ou método.

Existem quatro modos de visibilidade:

- ▶ Visibilidade privada: representada pelo símbolo -
- ▶ Visibilidade pública: representada pelo símbolo +
- ▶ Visibilidade protegida: representada pelo símbolo #
- ▶ Visibilidade pacote: representada pelo símbolo ~

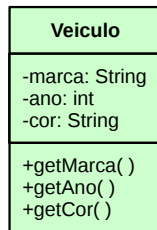
Veiculo
-marca -ano -cor
+getMarca() +getAno() +getCor()

Tipos dos Atributos

Podemos representar também no diagrama de classes o tipo de um atributo.

O tipo do atributo:

- ▶ É representado colocando-se **:tipo** após o nome do atributo.
 - ▶ O **tipo** pode ser primitivo (int, double etc.) ou se referir a uma classe.

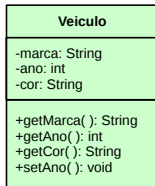


Métodos

O tipo de retorno

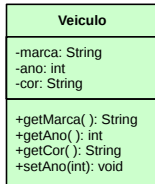
Podemos representar também no diagrama de classes o **tipo de retorno** de um método.

- ▶ É representado colocando-se **:tipo** após o nome do método.
 - ▶ O **tipo** pode ser primitivo (int, double etc.) ou se referir a uma classe.



Apesar de opcionais, os parâmetros também podem ser apresentados.

- ▶ Geralmente eles são representados de forma resumida.
 - ▶ Ou seja, mostrando apenas os seus tipos.



Nível de Detalhe



A UML é flexível, permitindo que você apresente a quantidade de detalhes que julgar necessário.



No diagrama de classes, por exemplo, apenas a **parte superior** do retângulo (nome da classe) é **obrigatória**. As demais são opcionais.

Em **grandes sistemas**, geralmente têm-se uma **versão** do diagrama de classes com uma **visão mais geral**, ou seja, contendo apenas o **nome** das classes e o **relacionamento** entre elas.

Mesmo no caso de diagramas mais detalhados, para **classes** contendo **muito membros**, é comum se **apresentar** apenas os **atributos e métodos principais**. Usa-se o ... para indicar que existem mais atributos e/ou métodos.



Diagrama de Classes - Outros Detalhes

O diagrama de classes pode ainda exibir outros detalhes¹, tais como:

- ▶ Valor inicial do atributo.
- ▶ Multiplicidade do atributo.
- ▶ Se o atributo é derivado (valor produzido por meio de algum cálculo).
- ▶ E outros ...

¹Esses detalhes são utilizados com menos frequência e estão fora do escopo desta disciplina.

Relacionamentos

Os relacionamentos entre as classes também são representados no Diagrama de Classes.

- ▶ Basicamente, os relacionamentos são representados por linhas ligando as classes envolvidas, com símbolo(s) na(s) ponta(s) para representar o **tipo de relacionamento**.

Mas quais são mesmo os **tipos de relacionamentos** possíveis entre classes?

- ▶ Composição.
- ▶ Agregação.
- ▶ Associação.
- ▶ Herança.



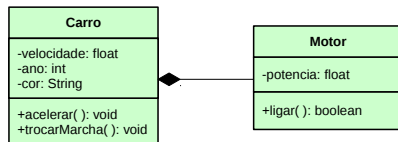
Relacionamentos: Composição

Vamos lembrar...

- ▶ O que caracteriza uma relação de **composição**?
 - ▶ Ela é uma relação do tipo todo/parte, onde o **todo** **controla** o tempo de vida da **parte**.

Como é a sua representação em um Diagrama de Classes?

- ▶ É feita com um losango preenchido na extremidade ligada à classe que representa o **todo**.



Atenção!

Observe que o atributo motor **não é representado** na divisão de atributos do retângulo que representa a classe **Carro**. Ou seja, a linha que representa o relacionamento já indica que **motor** é atributo de **Carro**.



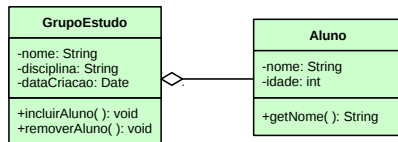
Relacionamentos: Agregação

Vamos lembrar...

- ▶ O que caracteriza uma relação de **agregação**?
 - ▶ Ela é uma relação do tipo todo/parte, onde o **todo não controla** o tempo de vida da **parte**.

Como é a sua representação em um Diagrama de Classes?

- ▶ É feita com um losango **vazado** na extremidade ligada à classe que representa o **todo**.



Atenção!

A mesma observação feita anteriormente vale aqui, ou seja, a linha que representa o relacionamento já indica que **aluno** é atributo de **GrupoEstudo**.

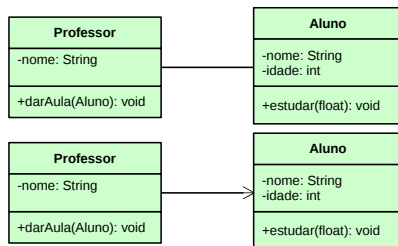
Relacionamentos: Associação

Vamos lembrar...

- ▶ O que caracteriza uma relação de **associação**?
 - ▶ A associação ocorre quando duas classes se relacionam, mas não representam um esquema todo/parte.

Como é a sua representação em um Diagrama de Classes?

- ▶ É feita apenas com uma linha ligando as duas classes.
- ▶ Opcionalmente pode-se usar uma seta no sentido da chamada dos métodos.



Atenção!

Vale a mesma observação feita anteriormente com relação ao atributo que representa o relacionamento.

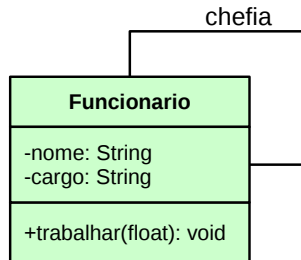
Relacionamentos: Associação

Temos também a associação **unária** ou **reflexiva**.

- ▶ O que caracteriza uma relação de **associação unária**?
 - ▶ Ela ocorre quando existe um relacionamento de um objeto de uma classe com objetos da mesma classe.

Como é a sua representação em um Diagrama de Classes?

- ▶ É feita da mesma maneira que uma associação qualquer.
- ▶ Veja no exemplo ao lado, que um funcionário pode ser **chefe** de outros funcionários.

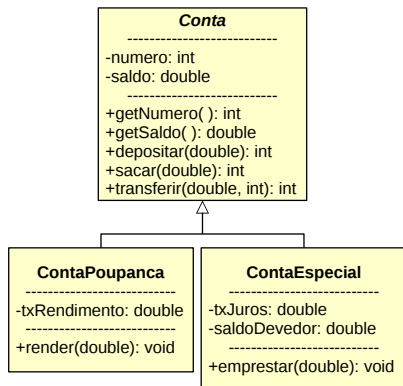


Relacionamentos: Herança

Vamos lembrar...

- ▶ O que caracteriza uma **herança** entre classes?
 - ▶ A herança permite que uma **subclasse** **herde** atributos e métodos de uma **superclasse**.

Como é a sua representação em um Diagrama de Classes?



- ▶ É representada por um **triângulo vazado** do lado da **superclasse**.
- ▶ Vale observar que a UML chama a herança de **generalização/especialização**.

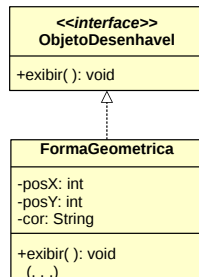
Relacionamentos: Interfaces

Será que a **herança de interfaces** é representada da mesma maneira que a **herança de classe**?

- ▶ Não é a mesma. A diferença é que para **interfaces** utilizamos a **linha tracejada**.

Na UML a **herança de interface** também tem um nome diferente: chama-se **realização**.

- ▶ Observe que utilizamos um **estereótipo** (<<NNN>>) para representar a interface.
- ▶ **Estereótipos** são uma maneira de **destacar** determinados componentes do diagrama.

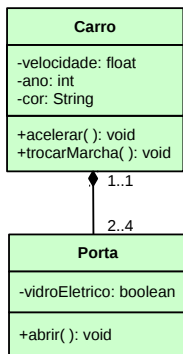


Multiplicidade

Uma outra informação pode ser adicionada aos nossos diagramas de classe: **multiplicidade**.

- ▶ Ela serve para determinar o número **mínimo** e **máximo** de **objetos** envolvidos em **cada extremidade** de um **relacionamento**.

Como é a sua representação em um Diagrama de Classes?



- ▶ É feita adicionando-se os **números mínimo e máximo** de objetos em **cada extremidade** da linha que representa o relacionamento.
- ▶ Veja no exemplo ao lado que:
 - ▶ Um carro pode ter de **duas** a **quatro** portas.
 - ▶ Uma porta pertence a somente **um** carro.

Multiplicidade

Seguem alguns exemplos de multiplicidade que podem ser utilizados em um relacionamento:

- ▶ 0..1 - no mínimo **zero** e no máximo **um**.
 - ▶ Ex.: Homem 0..1 — 0..1 Mulher
- ▶ 1..1 - no mínimo **um** e no máximo **um**².
 - ▶ Ex.: Carro 1..1 — 1..1 Motor
- ▶ 0..* - no mínimo **zero** e no máximo **vários**.
 - ▶ Ex.: Pessoa — 0..* Conta Bancária
- ▶ 1..* - no mínimo **um** e no máximo **vários**.
 - ▶ Ex.: Universidade — 1..* Departamento
- ▶ 4..6 - no mínimo **quatro** e no máximo **seis**.
 - ▶ Ex.: Caminhão — 4..6 Pneus

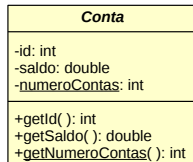
Vale observar que a multiplicidade acaba representando também as dependências entre os objetos.

²A multiplicidade 1..1 não precisa ser representada, uma vez que, quando a multiplicidade não está explícita, entende-se que é 1..1.

Atributos e Métodos Estáticos

Como eles são representados em um Diagrama de Classes?

- ▶ Isso é feito sublinhando-se o nome do atributo ou do método.



Outras possibilidades...

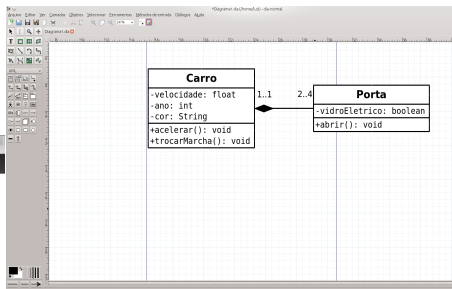
- ▶ A UML define várias outras representações no diagrama de classes.
 - ▶ Por exemplo, nas linhas que representam os relacionamentos podemos colocar **nomes** que auxiliem na **compreensão do tipo de vínculo** estabelecido entre os objetos das classes envolvidas.

Diagrama de Classes

Resumindo...

- ▶ Um diagrama de classes UML **facilita** a **representação das classes**.
- ▶ O uso de diagramas de classes **melhora a comunicação** entre **equipes** nas empresas.
- ▶ O diagrama de classes nos **permite entender** rapidamente a **modelagem do sistema**.

Editores de Diagramas



Perguntas?

